DOCUMENT RESUME

ED 212 647 TM 810 951

AUTHOR Morland, Richard B.
TITLE SPSS Beginner's Handbook for the Interactive Computer.
PUB DATE Sep 81
NOTE 17p.

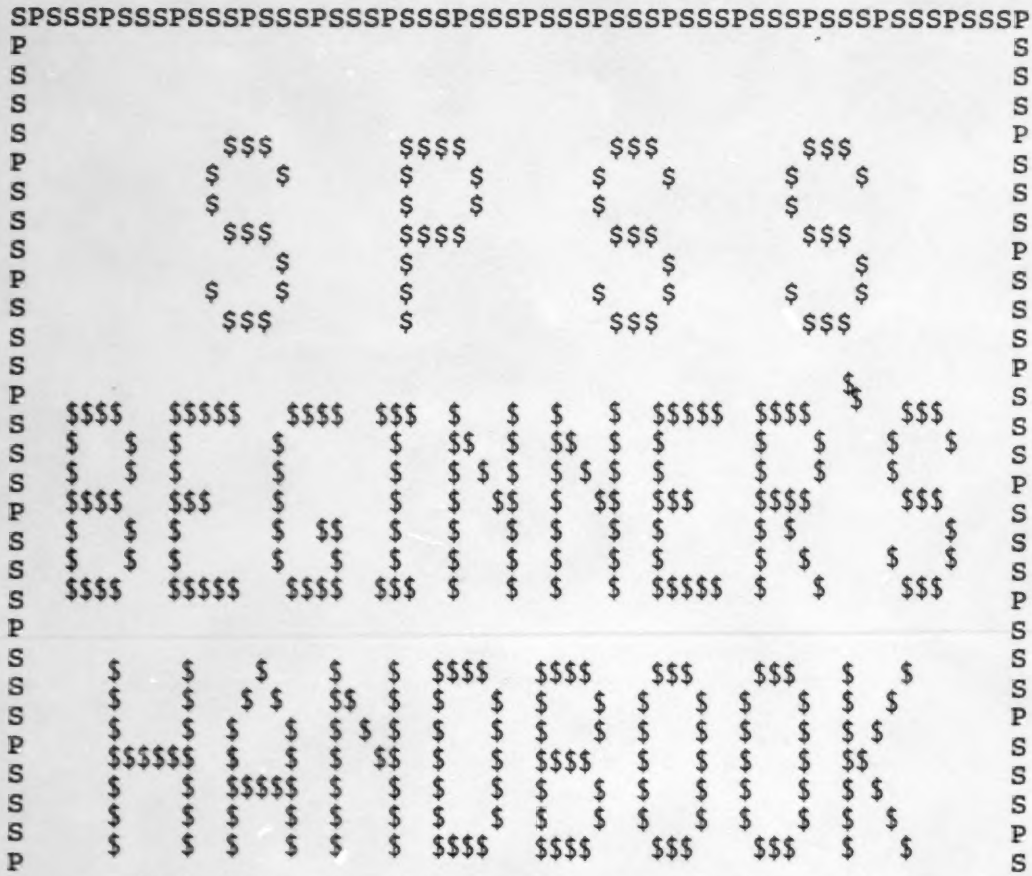
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Programs; *Data Processing; Programing Languages; Research Tools; Statistical Analysis
IDENTIFIERS *Interactive Computer Systems; *Statistical Package for the Social Sciences

ABSTRACT

This handbook lists step-by-step the procedures for making Statistical Package for the Social Sciences (SPSS) runs on the interactive computer. The programs follow the Loma Linda University Revision 2.1 as adapted for the Data General Eclipse Systems. The four-step process includes instructions for developing the codebook, building the data and command files in LINEDIT, and transferring the files to the SPSS disk. Two examples of common programs are presented. The first explains the procedures for the statistical analysis of survey data, while the second program gives the commands for deriving the statistical components of the regression equation for a predictive study. (Author/CE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED212647



FOR THE INTERACTIVE COMPUTER

(DATA GENERAL ECLIPSE SYSTEMS)

by

RICHARD B. MORLAND

Department of Education
Stetson University

#1 - September, 1981

SPSSSPSSSPSSSPSSSPSSSPSS

TM 810 951

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

R. B. Morland

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

INTRODUCTION

The purpose of this small manual is to introduce you to the SPSS system and to take you through a successful run. Once this is accomplished, you will be able to build on what you have learned to take full advantage of the inexhaustible capabilities of the system to analyze data.

What is SPSS? In a word, it is a computerized system for analyzing data with a speed that defies description. It performs the most complex multivariate analyses in a matter of seconds. Those of us who were brought up on the old rotary calculators had to spend untold hours solving a similar problem, and we were never sure our computations were correct. In addition to freeing you from the drudgery of repetitive statistical calculations, SPSS guarantees accuracy.

SPSS stands for the Statistical Package for the Social Sciences. The geniuses who developed the program are social scientists at the University of Chicago. Their text -- SPSS, 2nd ed., by Norman H. Nie et al., McGraw-Hill, 1975 -- is the source for you to consult when writing programs. A copy is on file in the Computer Center.

The SPSS system was designed originally for runs to be made with IBM punched cards instead of on the interactive computer, such as the AOS at Stetson. This explains why the references in the SPSS text are always to "cards." This fact is mentioned, because it is important for you to keep before you the mental image that you are actually typing your data onto an 80-column card instead of the lines displayed on the console screen. This means that if the screen jumps to a new line, you are still on the same line, insofar as the input to SPSS is concerned, provided that all of the characters, including blank spaces, do not exceed 80 columns.

A final word: The SPSS is most particular. Unless your instructions to the computer conform exactly to the SPSS format, the processing stops. To analogize, picture yourself at the console of a giant organ pecking out a tune. As soon as you hit a wrong note, the organ stops playing. This is what happens if there is anything in your input, even the omission of a single comma, to distort what SPSS expects. The run is aborted. Thus, it is imperative for your program to be free of mistakes. Seldom are first runs successful. Find the errors and try again. Assuaging the wrath of the computer, which in time begins to assume human characteristics, is worth the effort because of the incalculable benefits.

MAKING A SPSS ANALYSIS

A successful SPSS run is a four-part process: (1) Developing the codesheet; (2) Creating the raw data (XXX.DT) file; (3) Writing the program instructions in the command file (XXX.IN); and (4) Transferring your input to the SPSS disk. Each step is explained below.

Step 1 -- Building the Codesheet

The first task is to organize the data to be fed into the computer. This is done by hand on what is called a codebook or codesheet. You can purchase the blank sheets from the Bookstore or make one yourself using graph paper. What I prefer is to use one of the blank pages coughed up from print-out runs. An ample supply of these is available in the Computer Center. The rows are already numbered. Draw verticle lines about ¼ inch wide on the pages. This will give you 56 columns, which is more than adequate for most problems.

Enter your data in some logical fashion and in a form that it can be easily checked and edited. This is difficult if no spaces are left to break some of the columns. Accuracy is what you want here, for your study is no better than your data. For instance, if you enter 81 as the age of one of your high school students, instead of 18, the computer will accept your mistake. The computer does what it is told and trusts you to feed it correct information.

The example below is the coding of the responses to a 22-item questionnaire completed by 95 attorneys (see p. 8). The first three questions asked for demographic information. Each of the possible four responses to these questions required one column. Q4 is a two-column variable for the number of years practicing law; Q5 is a single column response to indicate the size of the law firm; Q6 (two columns) asked for the number of family cases handled each month. The remaining 16 questions were single response items. The codesheet looks like this:

Q	1-3			4		5	6		7 – 11					12 – 17						18 – 22				
1	2	1	9	1	7	3	2	0	2	2	2	1	2	1	1	2	2	2	1	2	2	1	2	3
2	2	1	2	0	9	2	0	0	3	2	1	2	5	1	1	9	2	1	2	1	3	1	4	1
3	2	1	3	1	0	2	0	6	3	3	2	3	5	2	2	2	2	2	2	2	5	2	4	1
.																								
.																								
95	2	9	1	1	5	4	0	2	5	5	2	5	6	1	2	3	2	3	1	1	2	2	1	3

Obviously, the width of the columns is reduced in this example. The Q at the head of each column designates the particular question.

If some subjects do not respond to a particular question, code these with a 9. Do not use the hyphen, zero, or leave blank. Where two columns are provided, as in Q4, and the response is less than 10, write these 06, 08, etc.

LINEDIT

With the codesheet developed and checked, you are now ready to type your data into the computer. First, a word about the Stetson system. It is the Advanced Operating System (AOS) manufactured by the DATA GENERAL CORPORATION. The Command Line Interpreter (CLI) is the program you will be in when you first log on. The CLI prompt is:). Once you are given the), you can move on to the other programs or directories.

The consoles in the Computer Center and elsewhere on campus vary somewhat, because they are not made by the same manufacturer. All have the standard typewriter keyboard, and most have individual keypads with an additional set of numbers for convenience. Both sets of numbers perform the same functions. The "cursor" is the blinking dash on the screen that indicates the point where your next input will fall. The two cursor keys move the position of input to the place on the line where you want it to appear. It is not necessary to first erase characters you want to replace. Just type the new words or numbers over them. The space bar erases; the rub-out key performs the same function in reverse.

All of your work in building your two SPSS files will be in LINEDIT, which has the prompt ?. Before you begin, familiarize yourself with some of its features. The commands you should know are:

SET LINE	Numbers all lines for ready reference
APPEND	To start or to add text to existing programs
LIST	Displays on screen those lines you command. LIST ALL displays everything in your program; LIST 1 5 displays lines 1 through 5.
MOD	Displays line or lines to be modified. MOD 5 8 will display in order lines 5 through 8; MOD 26 will show only line 26.
INS	Used to insert new lines and renumbers all lines following. NOTE: Once you invoke the MOD or INS commands, everything you type will become a part of your program. After you have completed editing, hit the ESC key. This will return you to the LINEDIT prompt.
DELE	To delete existing lines. DELE 3 5 will delete lines

3, 4, and 5. DELE 1 deletes the first line. In all instances where lines are removed, the existing lines are renumbered in order.

A knowledge of the double-key functions will be quite helpful when editing your programs. The control functions described below apply to the Data General consoles. The operations may be different on the other terminals.

In each of the following, first depress the CONTROL KEY (CTRL), and while holding it down, depress the second key.

CTRL-S Stops the display of information on the screen

CTRL-Q Restarts the display. NOTE: If CTRL-S is used, the computer will not receive input until CTRL-Q is hit.

CTRL-H Moves cursor to beginning of line

CTRL-A Moves cursor to end of line

CTRL-F Moves cursor to the next word down the line, word-by-word (or to the next column of digits)

CTRL-K Deletes everything to the right of cursor

CTRL-U Deletes entire line

CTRL-L Clears the screen, but does not abort input

CTRL-E Opens the line for inserting new materials. Position cursor on point of new input. After making the insertion, hit the space bar and then the return key. The line will close up. NOTE: If you want to make additional insertions on the same line after CTRL-E is depressed, the entire line of type will move until you cancel by hitting space bar or CTRL-E again.

CTRL-C CTRL-B (While holding down CTRL key, depress C then B). Aborts the editing and returns system to CLI.

On most of the other terminals, the control key is labelled, CONTROL. In the examples that follow the double-key functions are described as CONT-S, CONT-Q, etc. These are the same as CTRL-S, CTRL-Q, etc.

Information on other LINEDIT features, as well as the description of other operations of the Data General system, is available in the blue books on open file in the Computer Center. Don't hesitate to consult them for reference.

Finally, if you are working on a console and are "stuck," type HELP. Depending on the particular program, the screen will display instructions as to ways you can receive help.

Step 2 -- Creating the Raw Data File (XXX.DT)

In each of the examples that follow, this format is used:

What the screen displays is underscored, e. g., Username, Name of file to edit?), ?, 1 2 3 etc.
Your instructions to the computer are in capital letters, e.g., EDUC, X LINEDIT, SET LINE
My comments are in regular type, and in some places, within parentheses
The explanations refer to the lines in the program, e. g., L-1, L-12, etc.

```
                Hit return key to log on
Username:  EDUC
Password:  GRAD
 Here the computer gives the message of the day and other info

) X LINEDIT
Name of file to edit?  XXX.DT
Do you want the file to be created?   YES
?  SET LINE
?  APPEND
1                (Type in your data.  At any point in your
2                input you can see what you have typed by
3                hitting the ESC (escape) key and typing
.                LIST ALL.  Resume input by typing APPEND.
.                This will give you your next line number)
.
(N)              (Hit ESC when finished with last line of
                 input.  This takes you back to the LIN-
                 EDIT prompt ?
?  BYE           (This signs you off LINEDIT and brings
                 you back to CLI ).
:Udd:dir:educ:XXX.DT being updated
)        At this point, if you want to sign off computer, type:
                 BYE
         If you want the screen to display your input, type:
                 TY XXX.DT
         If you want a hard copy to be printed, type:
                 QPR XXX.DT

) BYE            This will sign you off the computer.  Always
                    do this
```

.DT Explanations

Any recognized pair of username-password will work.

Always hit the return key after you have completed your commands on a particular line.

X LINEDIT means execute LINEDIT

Your file name can be up to four letters and numbers, but it must start with a letter and end in .DT. Examples: JOY.DT, B345.DT, or HAL7.DT. The computer will search its files and will let you know if another file with the same name exists.

If you invoke the commands LIST ALL or TY XXX.DT, be ready to stop the display with CONT-S if you have more than 20 lines in your program. Resume with CONT-Q.

If you have completed your .DT input, you definitely will want a hard copy to proof for errors. It is helpful here to have someone read aloud the computer print-out line-by-line while you check the data from your codesheet.

NOTES

Step 3 -- Creating the Command File (XXX.IN)

Example #1 - First Run

(This example is a survey conducted by Sue Kilbride of family attorneys in south Florida toward the need for programs of court-sponsored counseling in cases of divorce where minor children are involved. Ninety-five attorneys responded to her 22-item questionnaire)

Log onto computer with Username and Password

```
)   X LINEDIT
Name of file to edit?  SUE.IN
Do you want the file to be created?   YES
?   SET LINE
?   APPEND
1   1234567890123456
2   RUN NAME        SURVEY OF FAMILY ATTORNEYS
3   PAGESIZE        NOEJECT
4   FILE NAME       SUE.DT
5   VARIABLE LIST   V01 TO V22
6   INPUT MEDIUM    DISK
7   N OF CASES      95
8   INPUT FORMAT    FIXED(3F1.0,1X,F2.0,1X,F1.0,1X,F2.0,1X,5F1.0,
1X,6F1.0,1X,5F1.0)
9   MISSING VALUES ALL(9)
10  FREQUENCIES     GENERAL=ALL
11  READ INPUT DATA
12  FINISH                (Hit ESC key)
?   DELE 1
?   LIST ALL
?   BYE
:udu:dir:educ:sue.in being updated
)   QPR SUE.IN
)   BYE
```

Explanations

Your command file must end in .IN

L-1. It is absolutely essential for you to establish the 16th column. In the SPSS system, the first 15 columns constitute the "control field," and the remaining 65 columns the "specification field." Any word or character, even a comma, that is typed into the first15 columns which is not recognized as a "control card"

will abort the run. The 16 digits can be erased at any point after the specification field is established. NOTE: L-8 is what will appear on the screen. In SPSS, you are still on the same line.

L-2. Call your study whatever is appropriate. The limit is 64 characters, including blank spaces. The run name will appear on each page of the SPSS print-out.

L-4. The file name will be the .DT input the computer will read.

L-5. The VARIABLE LIST, in this example, is simply the number of variables (questions) on the questionnaire.

L-7. The total number of respondents. The number of cases must be the same as the number of the last row in your .DT file.

L-8. Writing the INPUT FORMAT is the most difficult part of programming. You must type in FIXED and enclose the format within parentheses. Commas must separate the entries. In this example,

3F1.0 instructs the computer to read three 1-column variables
1X skip the next column, which is blank
F2.0 read one 2-column variable
1X skip the next column (also blank)
F1.0 read a 1-column variable
1X skip the next column
F2.0 read one 2-column variable
1X skip the next column
5F1.0 read the next five variables of one column each
1X skip the next column
6F1.0 read the next 6 columns, each with one variable
1X skip the next column
5F1.0 read the final five columns, each with one variable

Lay out your format in a similar fashion and make certain it conforms to the columns and variables in your .DT file. Where no numerals precede the F, the computer reads it as 1.

L-9. This assumes that the numeral 9 is not a response to any question. If it is, another format is required.

L-10. This command will give you a print-out for each question, broken down by categories of responses.

DELE 1. Make certain you delete the line of 16 digits.

LIST ALL. Check the screen display for errors.

QPR SUE.IN This will give you a print-out of your program. Double-check for errors, especially the INPUT FORMAT, before dumping onto the SPSS disk. If you are satisfied, you are now ready to transfer both files to the SPSS disk.

Step 4 -- Transferring Your Files to the SPSS Disk

After you have carefully checked your .IN file for mistakes, log back onto the computer. When you get the CLI prompt), type:

```
) SPSS
Command File?  SUE.IN
Data File?     SUE.DT
Done!
```

(Your own file names, of course, would be used)

This tells you that your input is now on the SPSS disk. It will be run the same day, usually after 5:00 in the afternoon, or the next morning at 8 o'clock. Stop by the Computer Center after these times to pick up your copy.

EXAMPLE #1 -- SECOND RUN

ADDING VAR LABELS, VALUE LABELS, AND STATISTICAL COMMANDS

Your first run will give you absolute numbers and percentages for each response on the questionnaire, but no identification other than the variable number. In this revision you want to name and attach values to each question as well as to collapse some of the categories if there are two few responses.

Log onto the computer as usual

```
) X LINEDIT SUE.IN
? SET LINE
? LIST ALL     (On this command the screen will display the 11 lines
                in your .IN program. You want to insert the VAR
                LABELS after MISSING VALUES, which is line 8).
? INS 9
9  VAR LABELS      V01 AGE/
10                 V02 SEX/
11                 V03 COMMUNITY POPULATION/
12                 V04 NUMBER YEARS PRACTICING LAW/
13                 V05 DESCRIPTION OF LAW FIRM/
          (Continue until you have named each of your 22 variables)
30                 V22 WOULD PROGRAM BE AN ASSET?/
31 VALUE LABELS    V01 (1)UNDER 30 (2)30-39 (3)40-49 (4)50+/
32                 V02 (1)MALE (2)FEMALE/
33                 V03 (1)OVER 100,000 (2)50,000 TO 100,000 (3)UND
ER 50,000/
34                 V04 (1)LESS THAN 5 (2)5-10 (3)10-20 (4)20+/
35                 V05 (1)SELF (2)2-3 (3)4-7 (4)8+/
```

```
36                  V06 (1)LESS THAN 5 (2)5-10 (3)10-20 (4)20+/
.       (Continue the identification of the values
.            for each variable)
.
42                  V19 TO V21 (1)STRG AGRE(2)AGREE(3)DISAGRE(4)STRG
DISAGRE(5)NO OPN/
43                  V22 (1)YES (2)NO (3)UNCERTAIN/
44 RECODE           V01 (1=2)/
45 RECODE           V04 (1 THRU 4=1) (5 THRU 10=2) (11 THRU 20=3) (21
THRU HI=4)/
46          (Hit ESC key.  This will bring you back into your
            program to FREQUENCIES, which is now line 46)
? LIST 46 48
46 Frequencies      General=All
47 Read Input Data
48 Finish
? INS 48
48 CROSSTABS        TABLES=V02 BY V22
49 STATISTICS       ALL
50 CROSSTABS        TABLES=V05 BY V21
51 STATISTICS       1
52          (Hit ESC key to return to original program.  FINISH
            is now line 52)
? LIST ALL

            (The screen will display your entire program.  Stop
            with CONT-S; restart with CONT-Q.  Examine and cor-
            rect errors invoking the MOD command)
? BYE
Do You Want To Save Original File As A Backup File?  NO
          (If you answer YES, screen will display:
          Backup File Is: :UDD:EDUC:SUE.IN.BU)
:udd:dir:educ:SUE.IN being updated
) QPR SUE.IN
) BYE
```

Transfer your revised .IN file to the SPSS disk in the same manner as before. Log back onto the computer.

```
) SPSS
Command File?  SUE.IN
Data File?     SUE.DT
Done!
```

Explanations for Example #1 -- Second Run

Be absolutely sure that the names for your VAR (variable) LABELS and the VALUE LABELS start in column 16. This is easily determined, because the lines in your original program will still be on the screen. If in doubt, move the cursor and count the spaces. If you still are not sure, type in the 16 digits (as shown on p. 8), set the column position, then delete the line of 16 digits.

The VAR LABELS must be a single name -- 40 character limit

Each VALUE LABEL can be up to 20 characters, but the computer will not print any letters or numbers in excess of 65. This limit includes the V# and the imbedded blanks.

The numbers of the response options for the VALUE LABELS must be enclosed within parentheses and conform to the numbers in the respective column in the .DT file.

There must be either a comma or a space after the variable and value label numbers, e. g.: V01,AGE or V01 AGE. It makes no difference to the computer whether or not spaces are within the line where the specific values are designated. Cf. lines 42 and 43.

L-33, L-42, and L-45. As explained above, you are still on the same line in SPSS, irrespective of what the screen displays. Your hard copy will confirm this. Just make certain you have not exceeded the 80 column limit for the total number of text and spaces for a single card. Check by adding 15 to the total number of characters in the specification field (where the value and variable labels are assigned).

L-44. The RECODE command is used because the first run revealed that only one of the attorneys was under 30. This few a number will invalidate the chi square statistic. The new instruction (1=2) places this individual in the 30-39 group (as shown in line 31).

L-45. RECODE is used here to bring into workable categories the number of years practicing law (V04) for statistical analysis.

L-46. The revised run will give the new frequencies for all values that have been recoded.

The slashes / instruct the computer to go directly to the next card (nothing after the slash is read). The / is not required.

The statistics you want will depend on the nature of your problem. In the present illustration, chi square is the appropriate statistic because the focus of the study was to determine if there were significant differences in the attitudes of the lawyers toward various

aspects of a court-sponsored program. In the SPSS system, CROSSTABS analysis includes the chi square. Only two crosstabulation tables are called for in this example, but you could obtain the measure of association for all possible combinations of variables. Whatever you request depends on the data you need to treat your hypotheses.

L-49 will give you the printed contingency table and 14 different statistical analyses. You have no need for most of these. L-51 gives you the crosstabulation table but only the chi square statistic. The numeral 1 instructs the computer to print only this statistic, which is the chi square.

There is no need to save the original .IN for a backup. If, however, you had made substantive changes and want to keep the original for possible future use, type YES.

NOTES

EXAMPLE #2 -- PREDICTING MEASUREMENTS

This example is representative of that employed in the public schools. A guidance counselor wanted to develop a three-dimensional graph to predict the score a high school student would make on the senior test, based on various combinations of the score made on the 9th Grade Test and the grade point average (GPA) achieved in high school. Of particular value in counseling was the ability to predict, within certain limitations, the GPA the student had to maintain to score within the accepted range on the Twelfth Grade Test. The researcher, Mary Broadaway, collected complete data for 83 students to develop the regression equation.

The SPSS analysis follows the same steps as in the preceding example.

Step 1 -- Building the Codesheet

Each of the two tests consists of three-digit numbers, and the GPA is reported on a four-point scale with two decimals. Leaving blank spaces (to simplify editing) between the variables, the codesheet consists of three 3-digit columns. THE DECIMAL POINT IN THE GPA IS IGNORED. SPSS supplies the decimal point as commanded in the input format.

Enter the data for the three variables into your codesheet. Your completed copy will consist of 83 rows and three columns of three digits. Double check for accuracy in recording.

Step 2 -- Creating the .DT File

The same procedure is followed as in Example #1. Log on, execute LINEDIT, and name your file. Then,

```
? SET LINE
? APPEND
   1  494 105 400
   2  438 091 342
   3  437 084 350
   .
   .
   .
  83  026 061 150      (hit ESC key when finished with last line)
? LIST ALL
       (Stop the display at appropriate points, examine your
       data, and make corrections using the MOD procedure)
? BYE
:udd:dir:educ:bun.dt being updated
```

As in Example #1, get a computer print-out with the command: QPR and your file name, in this case BUN.DT.

If you are satisfied there are no mistakes, move to Step 3.

Step 3 -- Creating the Command File

Except for the statistics, there is only one major change from the first example. Log on and into LINEDIT.

```
) X LINEDIT
Name of file to edit? BUN.IN
Do you want the file to be created? YES
?  SET LINE
?  APPEND
1  1234567890123456
2  RUN NAME        PREDICTION OF 12TH GRADE TEST SCORE FROM 9TH GRAD
E TEST & GPA
3  PAGESIZE        NOEJECT
4  FILE NAME       BUN.DT
5  DATA LIST       FIXED/1 TWELFTH 1-3, NINTH 5-7, GPA 9-11
6  INPUT MEDIUM    DISK
7  N OF CASES      83
8  INPUT FORMAT    FIXED (F3.0,1X,F3.0,1X,F3.2)
9  CONDESCRIPTIVE TWELFTH,NINTH,GPA
10 STATISTICS      ALL
11 READ INPUT DATA
12 PEARSON CORR    TWELFTH,NINTH,GPA
13 SCATTERGRAM     NINTH WITH TWELFTH/NINTH WITH GPA/TWELFTH WITH GPA
14 T-TEST          PAIRS=TWELFTH WITH NINTH/TWELFTH WITH GPA/NINTH WI
TH WITH GPA
15 REGRESSION      VARIABLES=TWELFTH,NINTH,GPA/
16                 REGRESSION=TWELFTH WITH NINTH,GPA(2)
17 STATISTICS      ALL
18 FINISH                  (Hit ESC key)
?  DELE 1                  (Don't forget to do this!)
?  LIST ALL
        (Check for errors.  If you find any, get back into
        LINEDIT and correct them)
?  BYE
        (Get your hard copy in the usual way and sign off)
```

Step 4 -- Transferring Input to SPSS Disk

The same procedure as above. Log on, type: SPSS and give the names of the Command and Data files.

Explanations -- Example #2

L-2, L-14. Bear in mind that SPSS reads these as being on one line.

L-5. The primary difference between this example and #1 is the addition of the DATA LIST card in the control field. Under this alternative method for entering data, the variables and the columns they occupy are identified. This eliminates the need for the INPUT FORMAT card. In this instance, however, the INPUT FORMAT card is required, because one of the columns contains decimals. This is why this particular example was chosen. The input format tells the computer to read the first variable of three columns (F3.0), skip the next column (1X), read the second variable of three columns (F3.0), skip the next column, then read the third variable which also occupies three columns with two digits to the right of the decimal (F3.2). To repeat, if DATA LIST is used, it is not necessary to include the INPUT FORMAT card unless the data contain decimals.

L-12 to L-17. All of these statistics are not needed, for many are repetitive. Both the T-TEST and REGRESSION, for example, print out the Pearson correlations. They are shown in order that you will know how to program for the particular statistic you want. CONDESCRIPTIVE prints out the mean, variance, and other descriptive statistics. PEARSON CORR presents correlation matrices. (Incidentally, only 3 seconds of computer time were required to compute the three correlation coefficients and 4 seconds to do the three correlated t ratios). The SCATTERGRAM command results in three scatter plots with the points for each pair of scores. T-TEST gives nine statistics for each combination of the three different variables. REGRESSION computes all of the statistics for the linear prediction equation, including the r, multiple R, beta coefficients, beta weights, standard errors, and F ratios.

CONCLUDING COMMENT

This little booklet has barely scratched the surface of the functions the SPSS system can perform with data. My hope is that the handbook will serve as the means of bringing you to the point where you feel comfortable writing SPSS programs. Armed with this ability, new horizons are open to you in research.

Because there are still some bugs in the conversion of SPSS to the interactive computer, obtaining successful runs can be as frustrating as solving Rubik's Cube. There are ways, however, to whip the gremlins. Hang in there! Few experiences exceed the joy of a successful SPSS run.